

AD-A173 985

A PLOTTING LIBRARY FOR TEKTRONIX COMPATIBLE DEVICES(U)

1/1

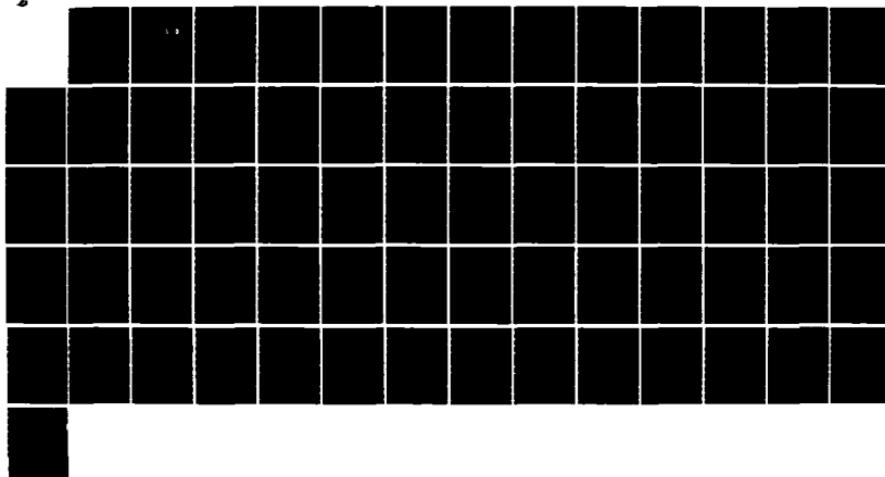
OPHIR CORP LAKWOOD CO J M HURMAN 09 AUG 85

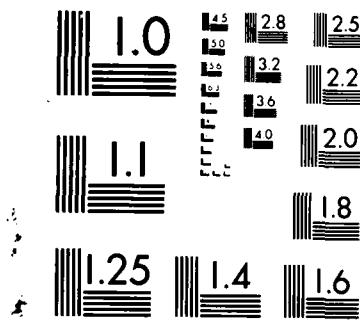
SCIENTIFIC-3 AFGL-TR-86-0014

UNCLASSIFIED

F/G 14/5

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

12
AD-A173 985

AFGL-TR-86-0014

A PLOTTING LIBRARY FOR
TEKTRONIX COMPATIBLE DEVICES

Joshua M. Wurman

OPHIR Corporation
7333 West Jefferson Avenue, Suite 210
Lakewood, CO 80235

DTIC
SELECTED
NOV 03 1986
S D
JMW

SCIENTIFIC REPORT NO. 3

8 August 1985

Approved for public release; distribution unlimited.

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

JMC FILE COPY

86 11 3 025

"This technical report has been reviewed and is approved for publication"

Barbara A. Main

BARBARA A. MAIN
Contract Manager

Arnold A. Barnes Jr.

ARNOLD A. BARNES, JR., Chief
Cloud Physics Branch

FOR THE COMMANDER

Robert A. McClatchey

ROBERT A. McCLATCHY, Director
Atmospheric Sciences Division

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify AFGL/DAA/LYC, Hanscom AFB, MA 01731. This will assist us in maintaining a current mailing list.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		Approved for Public Release; distribution unlimited	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFGL-TR-86-0014	
6a. NAME OF PERFORMING ORGANIZATION OPHIR Corporation	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Geophysics Laboratory	
6c. ADDRESS (City, State and ZIP Code) 7333 West Jefferson Avenue, Suite 210 Lakewood, CO 80235		7b. ADDRESS (City, State and ZIP Code) Hanscom AFB, Massachusetts 01731 Monitor/Barbara A. Main/LYC	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-83-C-0130	
9c. ADDRESS (City, State and ZIP Code)		10. SOURCE OF FUNDING NOS.	
		PROGRAM ELEMENT NO.	PROJECT NO.
		62101F	6670
		TASK NO.	WORK UNIT NO.
		12	CA
11. TITLE (Include Security Classification) A Plotting Library for Tektronix Compatible Devices	12. PERSONAL AUTHOR(S) Joshua Wurman		
13a. TYPE OF REPORT Sci. Rpt. No. 3	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) 8 August 85	15. PAGE COUNT 67
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Graphics, Tektronix, Plotting, Vax	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) This report contains documentation of a plotting library. This library contains FORTRAN callable subroutines that enable the creation of plots on Tektronix compatible terminals. This report includes a discussion of the library, a sample plotting session, and documentation of the routines.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Barbara A. Main		22b. TELEPHONE NUMBER (Include Area Code) (617) 377-2947	22c. OFFICE SYMBOL LYC

Table of Contents

I. Introduction	4
II. Definition of Terms	5
III. Operating Procedures	6
IV. Sample Plotting Session	7
V. Documentation of Routines in alphabetical order	8-67



Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail 3-d or Special
A-1	

I. Introduction

This report documents a plotting library for use with Tektronix terminals and emulators connected to Digital Equipment Company VAX computers using the VMS operating system. The library contains user callable routines that perform graphical manipulation on the screen, draw lines and markers, and contour data.

This work was conducted under contract with the Cloud Physics Branch, Atmospheric Sciences Division, of the Air Force Geophysics Laboratories

The routines in the library have been designed primarily for the Tektronix 4115B. Many of them will work on other terminals, but some, notably the routines that act on colors, the dialog area, and segments, may work only partially or not at all.

The library contains both higher and lower level routines. The higher level routines that are commonly called by user programs do things like contour arrays, draw axes, plot strings of text and change colors and visibilities of objects. Lower level routines control output and data encoding and are usually only called by the higher level routines in the library.

The most likely routines to be called by a high level user program are:

- T_BEG and T_END which must be called to initialize and terminate plotting,
- T_MARK and T_LINEM which plot markers and lines,
- T_BSEG, T_ESEG, T_PSEG, T_TSEG, T_VSEG, T_KSEG which begin, end, move, transform, set visibility, and kill graphical objects called segments,
- T_DACHAR, T_DAIND, T_DALIN, and T_DAPOS which set the dialog area width, color, number of lines, and position,
- T_LCHAR and T_SCHAR which change the dialog area character size,
- T_STRNG which plots a string of text,
- T_PANEL which draws a filled panel,
- T_PRINT which prints the screen,
- T_CONTOUR and T_AXES are powerful contouring and axes labelling routines. T_AXES is documented at the end of the alphabetical documentation

There are many other routines that are used primarily by other members of the library. These are documented here so that new routines that are written and added to the library can use them to do low level actions.

II. Definition of terms

Although familiarity with the Tektronix 4115B is assumed, some terms are defined here because they are used extensively throughout this documentation. If extensive use is made of the 4115, the 4110 Command Reference Manual, the 4115B Operators and Host Manuals should be consulted. The Command Reference Manual contains a comprehensive list of all valid commands as well as lists of colors, line styles, marker types, and panel fill patterns.

Dialog area: This is a plane that "floats" over the graphics on the screen. Data, output, and commands can be written to this plane without interfering with the graphics below.

Segments: A segment is a series of graphics commands or "primitives" that can be stored and manipulated as a whole.

Marker: This is a symbol like a plus sign, a diamond or a point. A list is contained in the 4110 Command Reference Manual

Panel: This is a polygon filled with a pattern or color. Patterns and colors are detailed in the 4100 Command Reference Manual

Alpha mode: In this mode, output is directed to the dialog area.

Vector mode: In this mode, output is directed to the graphics area.

III. Operating procedures

To use this plotting library, a program must be linked with the library, which now resides at DRA3:[WURMAN.TEK]TEKLIB.OLB. All routines and common blocks in this library have names that begin with the letters "T ". Thus, in order to eliminate naming conflicts, it is recommended that user routines and common blocks avoid names beginning with that sequence. Before running the program, the logical name PLOTIO must be defined. The definition of PLOTIO determines whether plots are sent to a file or to the screen. (But, note that the file output option is not fully functional in the current version of this library.) This is usually done with one of the following commands:

```
DEFINE PLOTIO TT      to send plots to the screen
or
DEFINE PLOTIO PLOT.DAT to send plots to a file
```

The first plotting routine called must be T_BEG and the last routine called must be T_END. It is not necessary to fully understand what they do, (they are discussed in the documentation) but, failure to call these routines may cause program execution to halt.

The terminal should be in alpha mode before any output is sent to the dialog area. Most commonly called library routines leave the terminal in that mode, but some of the lower level ones such as T_VECT and T_OUT do not.

IV. Sample Plotting Session

The following is an example of a program and a run session to plot a box on the screen.

```
$ edt box.for          ;create program
*I
C      this program will make a box and store it in a segment
C
C      integer*4 x(5) / 100, 500, 500, 100, 100 / !arrays for
C      integer*4 y(5) / 100, 100, 500, 500, 100 / !box vertices
C
C      call t_beg          !initialize plotting
C      call t_bseg( 1 )    !open segment 1
C      call t_lnsty( 0 )    !line type=0 (solid lines)
C      call t_lnind( 4 )    !line index=4 (blue by default)
C      call t_linem( 5, x, y ) !draw lines for box
C      call t_eseg          !close segment
C      call t_end           !terminate plotting
C
^Z
*EXIT
$ FORTRAN BOX          ;compile program
$ LINK BOX,DRA3:[WURMAN.TEK]TKLIB/LIB ;link to plot library
$ DEFINE PLOTIO TT        ;send output to screen
$ RUN BOX                ;run program to make plot
```

V. Documentation of Routines

The following pages contain a description of all routines in the library, with the exception of T_AXES (documented last, on page 65), in alphabetical order.

TYPE: SUBROUTINE

NAME: T_ALPHA

PURPOSE: To set a Tektronix terminal to alpha mode

DESCRIPTION: This subroutine will set a Tektronix terminal to alpha mode. The terminal mode is changed immediately.

CALLING SEQUENCE: CALL T_ALPHA

INPUT: none

OUTPUT: Calls T_OUT to send appropriate sequence to terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: All pending plotting output being stored by T_OUT will be sent to the terminal.

RESTRICTIONS: none

METHOD: An ASCII 31 is sent to the terminal with a call to T_OUT. The output is forced to be immediate by using a negative output length specifier.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman
Modified 7-Aug-1985 by Joshua Wurman
sends a -1 length to T_OUT
to force immediate output

TYPE: SUBROUTINE
NAME: T_ALPHAD
PURPOSE: To set a Tektronix terminal to alpha mode without forcing immediate output
DESCRIPTION: This subroutine will set a Tektronix terminal to alpha mode.
CALLING SEQUENCE: CALL T_ALPHAD
INPUT: none
OUTPUT: Calls T_OUT to send appropriate sequence to terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: An ASCII 31 is sent to the terminal with a call to T_OUT.
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1985 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_BEG

PURPOSE: Initializes plotting

DESCRIPTION: This subroutine assigns an I/O channel to the terminal or opens a file to receive plotting output and places information in the COMMON BLOCK /T_COMMON_CHAN/ concerning the I/O channel number and the action taken.

If the first two characters of the translation of the logical name PLOTIO are "TT" then an I/O channel is opened to the terminal and plots are sent there. If they equal anything else, then a file with the name of the translation of PLOTIO is opened and plots are sent there.

In addition, information about the action taken above is placed in a common block. This common block is used by other plotting output routines and is described in more detail below.

CALLING SEQUENCE: CALL T_BEG

INPUT: none

OUTPUT: A message indicating the action taken is sent to the terminal.

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: /T_COMMON_CHAN/ contains a two byte integer which contains either the channel number assigned to the terminal or a 50 which is the logical unit number associated with the file that is opened. A third byte contains either a .TRUE. if an I/O channel was opened to the terminal or a .FALSE. if a file was opened.

SIDE EFFECTS: If an error occurs during logical name translation, channel assignment, or file opening, program execution stops.

RESTRICTIONS: Since only the first two characters of the translation of PLOTIO are checked, it is impossible to send plots to a file with a name that starts with "TT...".

METHOD: The system service TRNLOG is called to translate "PLOTIO". If the translation starts with "TT" then the system service ASSIGN is called to assign the first available I/O channel. If the translation begins with anything else, then a file is opened with a LUN=50, CARRIAGECONTROL=LIST and RECL=1600.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY:

Created	7-Nov-1984	by Joshua Wurman
Modified	12-Nov-1984	by Joshua Wurman
		Send message to terminal telling
		of action
	19-Nov-1984	by Joshua Wurman
		Opens file if beginning of PLOTIO
		is not "TT" places .FALSE. in 3rd
		byte of /CHAN/ if file and .TRUE.
		if terminal
	29-Nov-1984	by Joshua Wurman
		Program stops if there is an error
		opening the file

TYPE: SUBROUTINE
NAME: T_BHSEG
PURPOSE: Closes the current segment and opens the next higher one
DESCRIPTION: Closes the segment that is currently open and opens the one that is next higher in number.
CALLING SEQUENCE: CALL T_BHSEG
INPUT: none
OUTPUT: Calls T_OUT to send appropriate sequence to terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Sends the "begin-next-higher-segment" command to the terminal.
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_BSEG

PURPOSE: Opens a specified segment

DESCRIPTION: Opens the segment specified in the calling argument

CALLING SEQUENCE: CALL T_BSEG(SEG_NUMBER)

SEG_NUMBER: Integer indicating which segment number
is to be opened

INPUT: none

OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequence to terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Sends "begin segment" command to terminal with appropriate
number code for the segment number by calling T_OUT and
T_INCOD

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE
NAME: T_COLM
PURPOSE: Changes an entry in the color map of the terminal
DESCRIPTION: Sets the hue, lightness and saturation of a specified color number to the specified values
CALLING SEQUENCE: CALL T_COLM(COL_NUM, COL_HUE, COL_LIGHT, COL_SAT)
COL_NUM: Integer containing the number of the color to be changed
COL_HUE: Integers containing the hue, lightness,
COL_LIGHT: and saturation to be associated with
COL_SAT: the color number
INPUT: none
OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to the terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: The color specifying mode is changed to HLS.
RESTRICTIONS: none
METHOD: T_OUT and T_INCOD are called to send appropriate sequences to the terminal
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE
 NAME: T_CONTOUR
 PURPOSE: To contour filled data arrays
 DESCRIPTION: This routine is a powerful color contouring routine. The caller has extensive control over the contour output. Most of the parameters that are passed to this routine are contained in a control file described below.

CALLING SEQUENCE: CALL T_CONTOUR (FIRST_SEGMENT , X_DIMENSION , Y_DIMENSION , EXIT_STATUS)

FIRST_SEGMENT: Integer containing the number of the first segment to be used in the plot. This segment contains the contours. The next higher numbered segment contains the extremum labels, if any.

X_DIMENSION: Integers containing the dimensions of the array to be contoured
 Y_DIMENSION:

EXIT_STATUS: Integer that will contain a one if this routine exits successfully or a minus one if there is an error exit

In addition, a subroutine, T_CONTOUR_INOUT, should be linked to the program. This routine is called to allow for boundary criteria to be specified. A sample routine is shown below. This sample will always return a .FALSE. which has no effect on T_CONTOUR. If a .TRUE. is returned, then contours will not be drawn in the grid point specified.

```

C Sample T_CONTOUR_INOUT that always returns an in bounds indication
C
C SUBROUTINE T_CONTOUR_INOUT ( RI , RJ , OUT )
C
C RI, RJ are four byte floating point numbers which contain a grid
C point location that corresponds to data gridpoint
C (RI+1,RJ+1)
C OUT is a one byte logical value which should be .TRUE. for
C out of bounds points and .FALSE. for in bounds points.
C
C LOGICAL OUT
C OUT=.FALSE.
C
C This block of code would leave a blank space for a label
C
C IF (RI.LT.30.AND.RI.GT.20.AND.RJ.LT.30.AND.RJ.GT.20)
C 1 THEN
C     OUT=.TRUE.
  
```

```

C      ELSE
C          OUT=.FALSE.
C      END IF
C
C      RETURN
C      END

```

INPUT:

Most parameters for the plot are read from the file with the logical name CONTOURCTL which should be defined prior to calling this routine. This can be done with a command like, DEFINE CONTOURCTL CONTROL.DAT. The file should have carriage return record attributes (which is what EDT creates). All the data in the file (except for one number) has a format of I5, F11.4, or A1. All data, except for the arrayed numbers, begin in the fourteenth column. The first two records in the file are ignored and can be used for labelling, formatting, etc.

An annotated copy of a sample control file follows. Note that any annotation line that begins with a ! should not actually appear in the file. Only the numbered lines 1-39 should appear in a usable control file.

```

!
!
! This file is a sample control file for T_CONTOUR
2 5678901234567890 I5 F11.4 A1
3MARK +0000 !Only used for debugging: places marks on contours
4IBW +0001 !Should be one: in future will disable segments...
5MESH +0004 !sub-grid scale for smoothing (1-9) 4 is good
!           !consumes program space: see restrictions
6LOGFAC +0000 !Base ten logarithmic factor applied to data
7NNC16 +0004 !Maximum of contours that can be plotted/16
!           !consumes program space: see restrictions
8CLABFRQ +0002 !Every CLABFRQ contour will be labelled
9CINT +00200.0000 !Contour interval: if neg then zero cont omitted
10ODDMAX +04000.0000 !Only every other cont over ODDMAX plotted
!           !Ignored if ODDMAX is zero
11CONLOW -00500.0000 !The lowest contour that can be plotted
12CONHIGH +04000.0000 !The highest contour that can be plotted
13FLAT +00000.0000 !Flat region contour supressing param (0.0-1.0)
!           !if zero then no suppression
!           !if one then most contours are suppressed
14HILOTHR +00050.0000 !Only extreemums that are HILOTHR different
!           !than the nearest contour value are plotted
15BYTEL L !Character plotted at minumums
16BYTEH H !Character plotted at maximums
17HTCONTL +00020.0000 !Height of contour labels
18HTHIOL +00020.0000 !Height of extreemum value labels
19HTBYTEL +00020.0000 !Height of extreemum type (max, min) labels
19.5NDCONT +0000 !Number of digits after decimal in cont labels
!           !Set to +0360 for wind dir format (000-360)
20NDHILO +0000 !Number of digits after dec in extreemum labels
!           !Set to +0360 for wind dir format (000-360)
21NEXTCOL +0015 !Color of extreemum value labels

```

```

22NHILOCOL +0015      !Color of extremum type (max, min) labels
23NLBCOL   +0015      !Unused
24IPRECLB  +0002      !Precision of text used in labeling: usu two
25IBEGSEG  +0003      !Unused
26EPS     .0000177    !A small uncommon number: Note format
!          !Used to adjust contours slightly
27MAXERRS +0250      !Max num of contouring errors before error exit
!
!The CONT array below contains thirty contour level thresholds at which
!line color and styles can be changed according to the STYL and COLR
!arrays. Thus, in this example, the 2800 contour will use the 25th value
!in the STYL and COLR arrays to choose line style and color
!This array must be arranged in ascending order
28 CONT1   -00100.0000 +00000.0000 +00050.0000 +00100.0000 +00150.0000
29 CONT2   +00200.0000 +00250.0000 +00300.0000 +00350.0000 +00400.0000
30 CONT3   +00450.0000 +00500.0000 +00800.0000 +01000.0000 +01200.0000
31 CONT4   +01400.0000 +01600.0000 +01800.0000 +02000.0000 +02200.0000
32 CONT5   +02800.0000 +03000.0000 +03200.0000 +03400.0000 +03600.0000
33 CONT6   +03800.0000 +04000.0000 +04200.0000 +04400.0000 +04600.0000
!The STYL and COLR arrays correspond to the CONT values
!Thus the contours between the nth and n+1th CONT values are drawn with
!the nth color and style (The 2800 contour would be in style=0,color=35)
34 STYL1   +0001 +0000 +0000 +0000 +0000 +0000 +0001 +0001
35 STYL2   +0000 +0000 +0000 +0000 +0000 +0000 +0000 +0000
36 STYL3   +0000 +0000 +0000 +0000 +0000 +0000 +0000 +0000
37 COLR1   +0021 +0022 +0023 +0024 +0025 +0026 +0007 +0028 +0029 +0030
38 COLR2   +0031 +0032 +0033 +0034 +0035 +0036 +0037 +0038 +0039 +0040
39 COLR3   +0041 +0042 +0043 +0044 +0045 +0046 +0047 +0048 +0049 +0050
!
!      End of control file
!
OUTPUT:      T_CONTOUR produces a contour plot in the graphics
               area of the screen and messages about the plot status
               in the dialog area
ACCESS:      The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: /T_CONTOUR_SCREEN/ Contains 4 integers. The first two are
               the x,y location of the lower left hand
               corner of the plot. The next two
               numbers are the width and height of the
               plot
               /T_CONTOUR_FIELD/ The data array to be contoured is
               passed in this block. This block must
               contain (NX+2)*(NY+2) four byte
               floating point numbers. This is
               for extrapolation beyond the data
               boundaries that is performed. The data
               should be sent in a contiguous form with
               no gaps. It should not be centered in
               the array. It is recommended that it be
               passed in a linear array with extra
               space at the end. T CONTOUR will take
               care of centering in the array.
SIDE EFFECTS: Program execution may halt if passed array sizes don't
               agree with the passed sizes in NX, NY and NC16, or if

```

there is no routine T_CONTOUR_INOUT linked to the image.

All possible types of errors have not been examined. While there is some error checking, passing unreasonable values in the control file, or the call arguments can cause the program to make bad plots and possibly halt.

RESTRICTIONS: Data values that are very close to contour values will cause occasional errors in contouring.

METHOD: The array is searched for contours and they are traced to completion. Data values are sometimes altered if they are very near contour values to prevent endless spiraling of contours. A bit mask of contours that have passed through particular gridpoints is used to prevent redrawing of contours. Cubic spline interpolation is used to calculate a subgrid scale in order to create smoother contours.

Data is passed through common block /T_CONTOUR_FIELD/ and the control file with a logical name of CONTOURL and through calling arguments.

Several internal subroutines, entries and functions, T_CONTOUR_DRAW_I, T_CONTOUR_DRAW_C, T_CONTOUR_EXTRPL, T_CONTOUR_EXTRPL_FNC, T_CONTOUR_INT_I, T_CONTOUR_INTNEW, T_CONTOUR_INTOLD, T_NUMCHAR, and T_CONTOUR_SRCHDIR, T_C_SET_IC, T_C_SET_IG, T_C_SET_SC, T_C_SET_TC, and T_C_SET_CM are called to perform contour drawing, extrapolation, interpolation, contour building, formatting, masking, and searching

T_CONTOUR_INOUT is called to test for map boundaries

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY:

Created	1984 by Joshua Wurman from old version at MIT
Modified	1985 by Joshua Wurman Extensive modifications to substitute MACRO routines for low level functions enhance speed and reliability

TYPE: SUBROUTINE
NAME: T_CONT240
PURPOSE: To contour arrays and send output to a VT240
emulating a Tektronix
DESCRIPTION: See documentation on T_CONTOUR for further
description
CALLING SEQUENCE: CALL T_CONT240 (see T_CONTOUR for arguments)
METHOD: Calls T_LINE240 to draw lines
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1985 by Joshua Wurman

TYPE: SUBROUTINES, ENTRIES, FUNCTION

NAME: T_CONTOUR_DRAW entries: T_CONTOUR_DRAW_I
T_CONTOUR_DRAW_C

T_CONTOUR_EXTRPL function: T_CONTOUR_EXTRPL_FNC

T_CONTOUR_INT entries: T_CONTOUR_INT_I
T_CONTOUR_INTNEW
T_CONTOUR_INTOLD

T_CONTOUR_SRCHDIR

T_NUMCHR

PURPOSE: These routines are called by T_CONTOUR to
plot arrays. It is unlikely that they
would ever be called by another routine.
See the source code for documentation.
They are included in the T_CONTOUR source file.

LANGUAGE: VAX-11 FORTRAN

TYPE: SUBROUTINE, ENTRIES

NAME: T_C_SET entries: T_C_SET_IC
T_C_SET_IG
T_C_SET_SC
T_C_SET_TC
T_C_SET_CM

PURPOSE: These routines are called by T CONTOUR to plot arrays. It is unlikely that they would ever be called by another routine. Documentation is included in the source code. They are contained in the file TCONTMAC source code file.

LANGUAGE: VAX-11 MACRO

TYPE: SUBROUTINE

NAME: T_DACHAR

PURPOSE: To set the dialog area characters per line

DESCRIPTION: Sets the dialog area characters per line to the specified value

CALLING SEQUENCE: CALL T_DACHAR(CHAR_PER_LINE)

CHAR_PER_LINE: The number of characters per line

INPUT: none

OUTPUT: Calls T_OUT, T_INCOD, AND T_DAVIS send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: Makes the dialog area visible

RESTRICTIONS: none

METHOD: T_OUT and T_INCOD are called to specify the characters per line. T_DAVIS is called with an argument of 1 in order to reset the dialog area and make the changes effective.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_DAIND

PURPOSE: Changes the color of the text and background in the dialog area

DESCRIPTION: Sets the foreground, background and erase indices of the dialog area.

CALLING SEQUENCE: CALL T_DAIND(FORE_COL, BACK_COL, ERASE_COL)

 FORE_COL: Integer containing the color index of the text in the dialog area

 BACK_COL: Integer containing the color index of the dialog area background

 ERASE_COL: Integer containing the color index in which characters are erased in the dialog area

INPUT: none

OUTPUT: Calls T_OUT, T_INCOD, and T_DAVIS to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: Makes the dialog area visible

RESTRICTIONS: none

METHOD: Calls T_OUT and T_INCOD to set the dialog area colors and then calls T_DAIND with an argument of 1 to reset the dialog area and make the changes effective

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_DALIN

PURPOSE: To set the number of lines in the dialog area

DESCRIPTION: Sets the number of lines in the dialog area to a specified value

CALLING SEQUENCE: CALL T_DAPOS(NUMBER_OF_LINES)

NUMBER_OF_LINES: Integer containing the number of dialog area lines

INPUT: none

OUTPUT: Calls T_OUT, T_INCOD, and T_DAVIS send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: Makes the dialog area visible

RESTRICTIONS: none

METHOD: T_OUT and T_INCOD are called to specify the number of dialog area lines. T_DAVIS is called with an argument of 1 in order to reset the dialog area and make the changes effective.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_DAPOS

PURPOSE: To set the position of the dialog area on the screen

DESCRIPTION: Sets the lower left hand corner of the dialog area to a specified location in screen coordinates

CALLING SEQUENCE: CALL T_DAPOS(X_POSIT, Y_POSIT)

X_POSIT: Integers containing the x and y position,
Y_POSIT: in screen coordinates, of the lower
- left hand corner of the dialog area

INPUT: none

OUTPUT: Calls T_OUT, T_XYCOD, and T_DAVIS send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: Makes the dialog area visible

RESTRICTIONS: none

METHOD: T_OUT and T_XYCOD are called to specify the position of the dialog area. T_DAVIS is called with an argument of 1 in order to reset the dialog area and make the changes effective.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_DAVIS

PURPOSE: To make the dialog area visible or invisible

DESCRIPTION: Sets the dialog area visibility to the specified value

CALLING SEQUENCE: CALL T_DAVIS (VISIBILITY)

VISIBILITY: Integer containing a 1 to set the dialog area visible or a 0 to set dialog area invisible

INPUT: none

OUTPUT: Calls T_OUT, and T_INCOD to send appropriate sequences to the Terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: If the dialog area is made visible, all pending changes to the dialog area parameters are made effective.

RESTRICTIONS: none

METHOD: T_OUT and T_INCOD are called to specify the number of dialog area lines.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_END

PURPOSE: Terminates plotting

DESCRIPTION: This routine sends all pending output to the device that is receiving plots and then either deassigns the I/O channel to the terminal or closes the plot file.

CALLING SEQUENCE: CALL T_END

INPUT: none

OUTPUT: Calls T_OUTNOW to cause all pending output to be sent

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: /T_CONTOUR_CHAN/ which is initialized in T_BEG
only the third byte is used by this subroutine

SIDE EFFECTS: If an error occurs during the deassignment of the I/O channel, program execution stops.

RESTRICTIONS: none

METHOD: The third byte in /CHAN/ is tested. If it is .TRUE., the system service DASSGN is called to deassign the I/O channel to the terminal. If it is .FALSE., then the plot file is closed and saved.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE
NAME: T_ESEG
PURPOSE: Closes the current segment
DESCRIPTION: Closes the segment that is currently open for definition
CALLING SEQUENCE: CALL T_ESEG
INPUT: none
OUTPUT: Calls T_OUT to send appropriate sequence to terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Sends "close segment" command to the terminal
by calling T_OUT
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE
NAME: T_GBCLR
PURPOSE: Changes the screen background color
DESCRIPTION: Sets the graphics background color to the specified color
CALLING SEQUENCE: CALL T_GBCLR(HUE , LIGHTNESS , SATURATION)
INPUT: none
OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to the terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Sends a "change background color" command with a call to T_OUT and then sends the three color parameters with subsequent calls to T_INCOD
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_INCOD

PURPOSE: Sends encoded integers to the terminal

DESCRIPTION: Encodes an integer into a one to three byte sequence and sends the code to the terminal

CALLING SEQUENCE: CALL T_INCOD(INTEGER)

INTEGER: Integer containing the integer to be encoded and sent to the terminal

INPUT: none

OUTPUT: Calls T_OUT times to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: Only integers in the range -32768 to +32767 inclusive will be encoded correctly in this routine.

METHOD: INTEGER is encoded according the the rules described in the Tektronix 4110 Series Command Reference manual. Integers with magnitudes less than 16 are sent as one byte; integers with magnitudes between 16 and 1023, inclusive, are sent as two bytes; integers with magnitudes greater than 1023 are sent as three bytes. T_OUT is called to send the correct sequence.

LANGUAGE: VAX-11 MACRO

MODIFICATION HISTORY: Created 7-Aug-1985 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_KSEG

PURPOSE: To erase segments permanently

DESCRIPTION: Deletes the specified segments from screen memory, making them permanently invisible and freeing screen memory for future segments

CALLING SEQUENCE: CALL T_KSEG (SEGMENT_NUMBER)

SEGMENT_NUMBER: Integer containing the number of the segment that is to be deleted. Note that if a -1 is passed, all segments will be erased.

INPUT: none

OUTPUT: T_INCOD and T_OUT are called to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Calls T_OUT and T_INCOD to send a "kill segment" command to the terminal

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_LCHAR

PURPOSE: To make dialog area text small

DESCRIPTION: Sets the alpha text writing mode to small

CALLING SEQUENCE: CALL T_LCHAR

INPUT: none

OUTPUT: Calls T_OUT and T_DAVIS to send appropriate sequence to terminal

ACCESS: The calling program must be linked to LIBT:TRLIB.OLS

COMMON BLOCKS: none

SIDE EFFECTS: Makes the dialog area visible
Empties buffer of all pending plot output .

RESTRICTIONS: none

METHOD: Calls T_OUT with a second parameter of -2 in order to send a two character sequence to the terminal immediately. T_DAVIS is called with an argument of one in order to cause an immediate effect on the text size.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman
Modified 9-Mar-1985 by Joshua Wurman
sends sequence through T_OUT with a negative length for immediate output

TYPE: SUBROUTINE

NAME: T_LINEM

PURPOSE: To draw a line on the terminal

DESCRIPTION: Connects a sequence of points, specified with (x,y) pairs, on the terminal

CALLING SEQUENCE: CALL T_LINE (NUMBER_OF_POINTS , X_ARRAY , Y_ARRAY)

NUMBER_OF_POINTS: Integer containing the number of points to be connected

X_ARRAY: Integer arrays containing the x and y

Y_ARRAY: coordinates of the points to be connected
The points should be in one to one correspondence

INPUT: none

OUTPUT: Calls T_OUT to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: Program execution may halt if an attempt is made to connect more than 1000 points in one call to this routine.
The terminal is left in alpha mode.

RESTRICTIONS: The maximum number of points that can be connected during one call to this routine is 1000. This limit is arbitrary and has been chosen to balance utility and use of space.

METHOD: Since the number of characters that can be sent in one output request is limited by factors that vary from machine to machine and from time to time, this routine sends a series of separate line draw requests. The number of requests is dependent on the maximum number of characters that can be sent at one time and the number of points to be connected. Currently the former quantity is explicitly set by a parameter statement in the routine.

To formulate each line draw request, the routine puts an "enter vector mode" request at the beginning of a buffer. Then it loops through the points, encoding them in a two to five byte format according to the conventions specified in the Tektronix 4110 Command Reference guide, and adds the codes to the buffer. (The variable length coding takes advantage of the fact that, if the highest or lowest order bits of the location of a point just drawn are the same as those of the current point, then those higher or lower order bits don't have to be sent for the current point.) This continues until the points are exhausted or the buffer reaches the maximum length that can be sent

with one output request. At that time an "enter vector mode" request is added to the buffer and T_OUT is called to send the buffer to the terminal. If more points remain then the process is repeated as necessary.

LANGUAGE: VAX-11 MACRO

MODIFICATION HISTORY: Created 1985 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_LNIND

PURPOSE: To change line color

DESCRIPTION: Sets the color in which future lines will be drawn
Does not affect lines already on the screen or in
memory

CALLING SEQUENCE: CALL T_LNIND (LINE_COLOR)

LINE_COLOR: Integer containing the line drawing
color

INPUT: none

OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences
to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Sends a "set line index" command with a call to T_OUT
and the specified line drawing color with a call to
T_INCOD

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_LNSTY

PURPOSE: To change line drawing style

DESCRIPTION: Sets the style in which future lines will be drawn
Does not affect lines already on the screen or in
memory

CALLING SEQUENCE: CALL T_LNIND (LINE_STYLE)

LINE_STYLE: Integer containing the line drawing
style

INPUT: none

OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences
to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Sends a "set line style" command with a call to T_OUT
and the specified line drawing style with a call to
T_INCOD

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_MARK

PURPOSE: To plot symbols

DESCRIPTION: Plots a specified number of markers in specified colors and locations

CALLING SEQUENCE: CALL T_MARK (X_ARRAY , Y_ARRAY , NUMBER , TYPE , COLOR)

X_ARRAY: Integer arrays containing the x,y locations
Y_ARRAY: of each symbol to be plotted

NUMBER: Integer containing the number of symbols to be plotted

TYPE: Integer containing the number code for the type of symbol to be plotted

COLOR: Integer containing the color in which the symbols are to be plotted

INPUT: none

OUTPUT: Calls T_OUT, T_LNIND, and T_XYCOD to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: Changes the line drawing color to the color specified for the plotted markers.
The terminal is left in alpha mode.

RESTRICTIONS: none

METHOD: This routine first sets the current marker plotting type with a call to T_OUT and T_INCOD. Then the color of the markers is set with a call to T_LNIND which, in addition to setting the line color, sets the marker color. Then the terminal is sent into marker plotting mode with a call to T_MMODE. Each x,y location passed in the X_LOCATION and Y_LOCATION arrays is encoded and sent to the terminal with calls to T_XYCOD. Finally, the terminal is set to alpha mode with a call to T_OUT.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman
Modified 7-Aug-1985 by Joshua Wurman
calls T_MMODE to enter marker mode

TYPE: SUBROUTINE

NAME: T_MMODE

PURPOSE: To set a Tektronix terminal to marker mode

DESCRIPTION: This subroutine will set a Tektronix terminal to marker mode. The terminal mode is changed immediately.

CALLING SEQUENCE: CALL T_MMODE

INPUT: none

OUTPUT: Calls T_OUT to send appropriate sequence to terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: All pending plotting output being stored by T_OUT will be sent to the terminal.

RESTRICTIONS: none

METHOD: An ASCII 28 is sent to the terminal with a call to T_OUT. The output is forced to be immediate by using a negative output length specifier.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 7-Aug-1985 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_NUMCHR

PURPOSE: Called by T_CONTOUR and T_AXES to plot arrays and draw axes. It encodes numbers into character format. It has not been tested outside of the limited use in the T_CONTOUR and T_AXES calls and is thus not documented here. See the T_CONTOUR source code for further documentation.

LANGUAGE: VAX-11 FORTRAN

TYPE: SUBROUTINE
NAME: T_MOVE
PURPOSE: To move the cursor without drawing a line
DESCRIPTION: Moves the beam location to the specified position
in a "pen up" mode
CALLING SEQUENCE: CALL T_MOVE (X_LOCATION , Y_LOCATION)
X_LOCATION: Integers containing x,y location
Y_LOCATION: of the target location
INPUT: none
OUTPUT: Calls T_OUT and T_XYCOD to send appropriate sequences to
the terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Sends an explicit "move" command to the terminal with
calls to T_OUT and T_XYCOD
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_OUT

PURPOSE: To send output

DESCRIPTION: T_OUT decides whether to add plot output to a buffer, or to send it output to a device. The decision is based on several factors:

- If the output device is the user's terminal, then all output is sent immediately.
- If the output length is passed as a negative value, all current and stored output is sent immediately
- If the output length is passed as a zero, all data currently in the buffer is sent immediately
- If the buffer is full, then the output is sent immediately.
- Otherwise, the output is appended to the buffer.

CALLING SEQUENCE: CALL T_OUT (OUTPUT , OUTPUT_LENGTH)

OUTPUT: Character variable containing the output to be sent

OUTPUT_LENGTH: Integer containing :
--the number of characters to be sent if buffered output allowed
--the negative of the number of characters to be output if all current and pending output is to be sent
--zero if all pending output is to be sent

INPUT: none

OUTPUT: Calls T_OG to send output if output is to be sent

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: /T_COMMON_BUF/ contains the output buffer and is shared with T_OG

/T_COMMON_CHAN/ contains a two byte integer which is ignored by T_OUT and a third byte which is .TRUE. if an I/O channel is opened to the terminal or .FALSE. if a file is open

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: If the output device is the terminal (ITT=.TRUE.) then the output buffer is filled with the input characters and T_OG is called.

TYPE: SUBROUTINE
NAME: T_OUTNOW
PURPOSE: To send all pending output
DESCRIPTION: This is the routine that programmers should use to finish plots etc. It causes the pending output buffer to be sent to the output device. If all plotting is to be ended, T_END should be called instead.
CALLING SEQUENCE: CALL T_OUTNOW
INPUT: none
OUTPUT: Calls T_OUT to send all pending output
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Calls T_OUT with a length argument of zero to force all pending output
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_PNCLR

PURPOSE: To set the panel filling color

DESCRIPTION: Sets the panel filling color or style

CALLING SEQUENCE: CALL T_PNCLR (COLOR)

COLOR: Integer containing the filling color or style of future panels

INPUT: none

OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Calls T_OUT to send a "set fill pattern" command to the terminal and then calls T_INCOD to send the fill pattern

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE
NAME: T_PREC
PURPOSE: To enable low or high precision text writing in the graphics area
DESCRIPTION: Sets the graphtext precision to string or stroke precision
CALLING SEQUENCE: CALL T_PREC (PRECISION)
PRECISION: Integer containing a one if string precision is desired or a two if stroke precision is desired.
INPUT: none
OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to the terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Calls T_OUT and T_INCOD to send a "set graphtext precision" command to the terminal with the specified parameter
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_PRINT

PURPOSE: To print the screen

DESCRIPTION: Causes a hardcopy of the screen to be produced with either a positive or negative background

CALLING SEQUENCE: CALL T_PRINT (BACKGROUND_CODE)

BACKGROUND_CODE: Integer containing a zero or a one if a normal hardcopy is desired and a two if a reversed background copy is desired

INPUT: none

OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to the terminal which, in turn causes a hardcopy to be produced

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Sends a "hardcopy" command with the appropriate parameter with calls to T_OUT and T_INCOD

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_PSEG

PURPOSE: To move objects on the screen

DESCRIPTION: This routine will move a specified segment's pivot point to the specified location.

CALLING SEQUENCE: CALL T_PSEG (SEGMENT_NUMBER , X_LOCATION , Y_LOCATION)

SEGMENT_NUMBER: Integer containing the number of the segment to be moved

X_LOCATION: Integers containing the x,y location
Y_LOCATION: to which the segment is to be moved

INPUT: none

OUTPUT: Calls T_OUT, T_INCOD, and T_XYCOD to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Sends a "set segment position" command to the terminal with calls to T_OUT, T_INCOD, and T_XYCOD

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_RECOD

PURPOSE: To encode real numbers and send them to the terminal

DESCRIPTION: Encodes a specified real number into valid mantissa and exponent integers that can be sent to the terminal

CALLING SEQUENCE: CALL T_RECOD (REAL_NUMBER)

REAL_NUMBER: Four byte real number containing the number to be sent to the terminal

INPUT: none

OUTPUT: Calls T INCOD to send the mantissa and exponent to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: This routine uses a scheme that closely parallels that described in the Tektronix 4110 Command Reference guide for encoding real numbers. The real number is broken down into a base-two integer mantissa and exponent and then both of these are sent to the terminal with calls to T_INCOD.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE
NAME: T_RFRSH
PURPOSE: Clears the screen
DESCRIPTION: This routine will clear the screen of all graphical objects that are not stored in segments
CALLING SEQUENCE: CALL T_RFRSH
INPUT: none
OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to the terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Calls T_OUT to send a "renew view" command to the terminal and calls T_INCOD to send an argument of zero
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman
Modified 7-Aug-1985 by Joshua Wurman
sends an argument of zero to specify the current view

TYPE: SUBROUTINE
NAME: T_ROT
PURPOSE: Sets the text rotation angle
DESCRIPTION: Sets the graphtext rotation angle for future graphtext output
CALLING SEQUENCE: CALL T_ROT (ANGLE)
ANGLE: Four byte floating point number containing the angle, in degrees, that future graphtext should be rotated
INPUT: none
OUTPUT: Calls T_OUT and T_RECOD to send appropriate sequences to the terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Calls T_OUT to send a "set graphtext rotation" command and T_RECOD to send the rotation angle
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE
NAME: T_SCHAR
PURPOSE: To make dialog area text large
DESCRIPTION: Sets the alpha text writing mode to large
CALLING SEQUENCE: CALL T_SCHAR
INPUT: none
OUTPUT: Calls T_OUT and T_DAVIS to send appropriate sequence to terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: Makes the dialog area visible
Empties buffer of all pending plot output
RESTRICTIONS: none
METHOD: Calls T_OUT with a second parameter of -2 in order
to send a two character sequence to the terminal
immediately. T_DAVIS is called with an argument of
one in order to cause an immediate effect on the text
size.
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman
Modified 20-Mar-1985 by Joshua Wurman
sends sequence through T_OUT with a
negative length for immediate output

TYPE: SUBROUTINE

NAME: T_STRING

PURPOSE: To plot strings of characters

DESCRIPTION: This routine will plot character strings of specified height, color, rotation, and precision at specified locations.

CALLING SEQUENCE: CALL T_STRNG (X_LOCATION , Y_LOCATION , HEIGHT , CHARACTER_STRING , ROTATION_ANGLE , NUMBER_OF_CHAR , PRECISION , COLOR)

X_LOCATION: Integers containing the x,y location

Y_LOCATION: of the lower left corner of the string

HEIGHT: Integers containing the height of the printed characters

CHARACTER_STRING: Character variable containing the character string

ROTATION_ANGLE: Four byte floating point number containing the angle in degrees that the string is to be rotated

NUMBER_OF_CHAR: Integer containing the number of characters passed in CHARACTER_STRING

PRECISION: Integer containing a one for string precision characters and a two for stroke precision characters

COLOR: Integer containing the color index of the characters

INPUT: none

OUTPUT: Calls T_VECT, T_MOVE, T_PREC, T_TSIZ, T_ROT, T_TXCLR, T_OUT, T_INCOD, and T_ALPHA to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: Leaves the terminal in alpha mode

RESTRICTIONS: none

METHOD: The active position is moved to the specified location with a call to T_MOVE, the text attributes are set with calls to T_PREC, T_TSIZ, T_ROT, and T_TXCLR, and then the character string is sent to the terminal using T_OUT.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_TBCLR

PURPOSE: To set the string precision text background and the color in dashed line gaps

DESCRIPTION: This will set the background that appears behind characters sent in string precision mode. It will also set the color that is drawn in the gaps of dashed lines.

CALLING SEQUENCE: CALL T_TBCLR (TEXT_BACKGROUND_COLOR ,
 DASHED_LINE_GAP_COLOR)

TEXT_BACKGROUND_COLOR: Integer containing the color index to be displayed behind string precision text

DASHED_LINE_GAP_COLOR: Integer containing the color index to be displayed in the gaps of dashed lines

INPUT: none

OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to the Terminal

ACCESS: The calling program must be linked to LIBF:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: This sends a "set background indices" command with calls to T_OUT. The parameters are sent with calls to T_INCOD.

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_TSEG

PURPOSE: To move and modify segments

DESCRIPTION: This will move, rotate and scale segments

CALLING SEQUENCE: CALL T_TSEG (SEGMENT_NUMBER , X_SCALE , Y_SCALE
ROTATION_ANGLE , X_LOCAT , Y_LOCAT)

SEGMENT_NUMBER: Integer containing the number
the segment to be altered

X_SCALE: Four byte floating point numbers containing
Y_SCALE: the x and y scaling factors. Numbers
greater than one stretch the segment,
numbers less one compress it.

ROTATION_ANGLE: Four byte floating point number
containing the angle through which
the segment is to be rotated

X_LOCAT: Integers containing the x and y location
Y_LOCAT: to which the pivot point of the segment
is to be moved

INPUT: none

OUTPUT: Calls T_OUT, T_INCOD, T_RECOD, and T_XYCOD to send
appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Sends a "set segment image transform" command to the
terminal will calls to T_OUT, T_INCOD, T_RECOD, and
T_XYCOD

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_TSIZ

PURPOSE: To set the graphtext size

DESCRIPTION: Sets the graphtext height, width, and spacing according to a passed height value

CALLING SEQUENCE: CALL T_TSIZ(CHARACTER_HEIGHT)

CHARACTER_HEIGHT: Integer containing the height of future graphtext characters. The width will be set to 4/5 of the height and the spacing between characters will be set to 1/5 of the height.

INPUT: none

OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Calls T_OUT and T_INCOD to send a "set graphtext size" command with parameters of 4/5, 1, and 1/5 of CHARACTER_HEIGHT

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_TXCLR

PURPOSE: To set text colors

DESCRIPTION: This will set the color index for future alphatext and graphtext

CALLING SEQUENCE: CALL T_TXCLR (COLOR)

COLOR: Integer containing the color index of future text

INPUT: none

OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: Sends a "set text index" command with calls to T_OUT and T_INCOD

LANGUAGE: VAX-11 FORTRAN

MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE
NAME: T_VECT
PURPOSE: To set the terminal to vector mode
DESCRIPTION: This will set and leave the terminal in vector mode
CALLING SEQUENCE: CALL T_VECT
INPUT: none
OUTPUT: Calls T_OUT to send appropriate character to the terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Sends an ASCII 29 (decimal) to the terminal with
a call to T_OUT
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE
NAME: T_VSEG
PURPOSE: To make segments visible and invisible
DESCRIPTION: This will set a segment's visibility to on or off
CALLING SEQUENCE: CALL T_VSEG (SEGMENT_NUMBER , VISIBILITY)
SEGMENT_NUMBER: Integer containing the number of
the segment to be made visible or
invisible
VISIBILITY: Integer containing a zero if the segment
is to be invisible or a one if the
segment is to be visible
INPUT: none
OUTPUT: Calls T_OUT and T_INCOD to send appropriate sequences to
the terminal
ACCESS: The calling program must be linked to LIBT:TKLIB.OLB
COMMON BLOCKS: none
SIDE EFFECTS: none
RESTRICTIONS: none
METHOD: Sends a "set segment visibility" command with calls to
T_OUT and T_INCOD
LANGUAGE: VAX-11 FORTRAN
MODIFICATION HISTORY: Created 1984 by Joshua Wurman

TYPE: SUBROUTINE

NAME: T_XYCOD

PURPOSE: To send a location to the terminal

DESCRIPTION: Sends an encoded x,y location to the terminal

CALLING SEQUENCE: CALL T_XYCOD (X_LOCATION , Y_LOCATION)

X_LOCATION: Integers containing the x,y
Y_LOCATION: coordinates of a point to be
sent to the terminal

INPUT: none

OUTPUT: Calls T_OUT to send the encoded location to the terminal

ACCESS: The calling program must be linked to LIBT:TKLIB.OLB

COMMON BLOCKS: none

SIDE EFFECTS: none

RESTRICTIONS: none

METHOD: This routine encodes the x,y location into a string of five ASCII characters according to the rules outlined in the Tektronix 4110 Command Reference guide and sends the five characters with T_OUT

LANGUAGE: VAX-11 MACRO

MODIFICATION HISTORY: Created 1985 by Joshua Wurman

TYPE: SUBROUTINE
NAME: T_AXES entries: T_AXES_GRAPH
T_AXES_GXFRM
PURPOSE: To draw labelled axes
DESCRIPTION: Subroutine T_AXES is useful for plotting a rectangle with labelled axes to surround the contour map. It can also be used to plot axes for drawing graphs and scatter diagrams etc.
CALLING SEQUENCE: CALL T_AXES (argument list shown below)

RX1	Minimum X value - left-hand side of X axis
RX2	Maximum X value - right-hand side
RDX	Step size between labelled X values
XFMT	Format string to use for X values This must include surrounding parentheses, and end with a zero byte (i.e., it is an ASCII zero string) For example - '(F4.2)' or '(I2)' ** First Special Case - if the string starts with an 'L' rather than '(' then the numbers RX1 etc. are treated as logs to the base 10, and 10^{**RX1} etc. are plotted out as the labels ** Second Special Case - if the string is just '(YM)' then RX1 must be a first year and month (e.g., 8204) and RX2 must be a final year and month (e.g., 8302). The axis will be labelled with APR, MAY, etc.
NTX	Number of ticks in X per labelled value (If NTX=1, only labelled values are ticked)
LDX	A logical variable which is .TRUE. if the labelled values should have leading zeroes plotted
WIDTH	The width in pixels of the base of the rectangle - i.e., the distance from RX1 to RX2. The bottom left corner of the rectangle will not be at the origin - it will be shifted in and up an appropriate distance to allow for the axis labelling
XLAB	Label for the X axis - an ASCII zero string
RY1	Minimum Y value - bottom of Y axis
RY2	Maximum X value - top of Y axis
RDY	Step size between labelled Y values
YFMT	Format string to use for Y values This must include surrounding parentheses, and end with a zero byte (i.e., it is an ASCII zero string) For example - '(F4.2)' or '(I2)' ** First Special Case - if the string starts with an 'L' rather than '(' then the numbers RY1 etc. are treated as logs to

the base 10, and 10**RY1 etc. are plotted
 out as the labels
 NTY Number of ticks in Y per labelled value
 (If NTY=1, only labelled values are ticked)
 LDY A logical variable which is .TRUE. if the
 labelled values should have leading zeroes
 plotted
 HEIGHT The height in pixels
 of the rectangle - i.e., the distance from
 RY1 to RY2.
 YLAB Label for the Y axis - an ASCII zero string
 CHHT Height (pixels) of the characters used
 for the axis labels and numbers
 This is also used to get an appropriate
 scale for the ticks etc.
 GRID A logical variable - if .TRUE., a complete
 grid is drawn at the X and Y labelled values
 TWOTIK A logical variable - if .TRUE., ticks are
 put on all 4 sides of the rectangle rather than
 just the bottom (X) and left (Y)
 BSTY Axis and tick line style
 BCOL Axis and tick line color
 XCOL X axis label color
 YCOL Y axis label color
 IBSEG beginning segment number: axes and labels
 are placed in different segments IBSEG and
 IBSEG+1
 IBW controls whether segments and color commands
 are made. If IBW=-1 then no segments and
 colors are made. This can be used for
 making plots on black and white terminals
 which do not support segment memory.

A sample call to T_AXES to simply plot X and Z each from 0 to
 50 in steps of 10, with 10 ticks per label (so that the ticks
 are spaced one unit apart) is:

```

    CALL T_AXES( 0.0,50.0,10.0,'(I2)',10,.FALSE.,2000.,'X',
1      -0.0,50.0,10.0,'(I2)',10,.FALSE.,2000.,'Y',
2      20...FALSE.,.TRUE.,1,0,1,1,1,0)
  
```

Note that the numbers labelling the axes are plotted in I2 for-
 mat and do not have leading zeroes; the rectangle is actually
 a 2000 pixel square; the character height is 0.15 inch (a pretty
 good value); a grid is not plotted, but ticks are put on all
 sides of the box.

T_AXES has an entry point T_AXES_GXFRM which can be used to
 transform an (X,Y) value in the ordinate/abscissa space to (X,Y)
 plot co-ordinates. For example, CALL
 T_AXES_GXFRM(RX1,RY1,XBOT,YLEFT) will return in XBOT the plot
 co-ordinate of the base of the rectangle, and in YLEFT the plot
 co-ordinate of the left of the triangle. Clearly, by doing a
 series of such calls for one can obtain points to be joined to
 draw a graph within the rectangle. There is also an entry point
 T_AXES_GRAPH (see the source program for details).

In the special case of logarithmic values in X or Y, the input number is treated as a log value; in the special case of years and months in X, the input value must be 1 for the first month (left-hand side), incremented by 1 for each successive month.

INPUT: none

OUTPUT: Calls `T_BSEG`, `T_ESEG`, `T_STRNG`, and `T_LINE` to create output on the terminal

ACCESS: The calling subroutine must be linked with LIBT:TKLIB.OLB

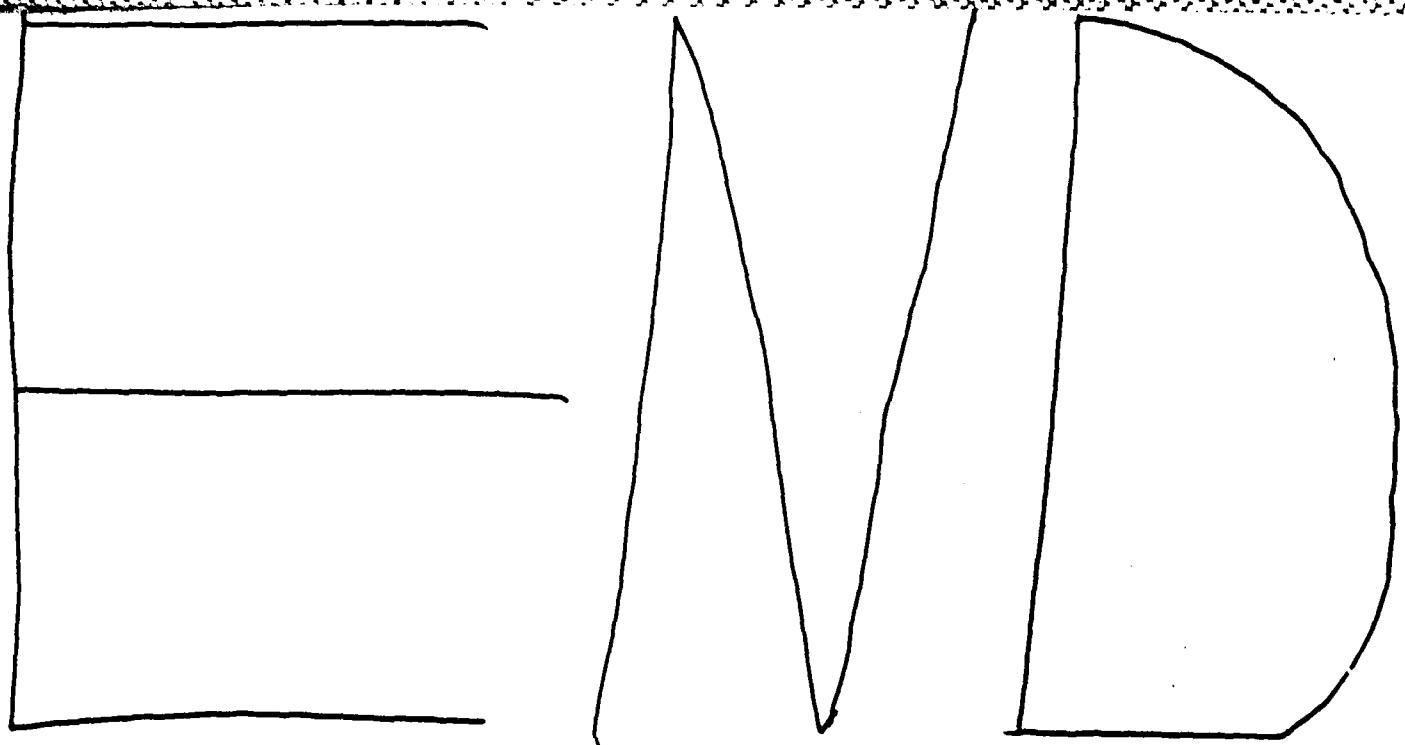
COMMON BLOCKS: none

SIDE EFFECTS: Terminal may be left in alpha mode

RESTRICTIONS: none

METHOD: see source code

LANGUAGE: VAX-11 FORTRAN



2-86

